ACST886_Assignment1

Chieh Cheng (44472587)

August 21, 2019

Q1

Initial Setttings:

```
#Amount Borrrowed:
Loan<-600000
#interest rate:
i<-0.035
#length of repayment:
n<-30
#First instalment is due on 19/03/2020
```

Suppose the annual repayment is x. a_n represents the annuity-immediate for n years.

$$x * a_n = Loan$$

In order to use "uniroot", I define the objective function "Q1" based on the discounted cash flow equation:

```
Q1<-function(x){x*(1-(1/(1+i))^n)/i - Loan}
(AnswerQ1<-uniroot(Q1,interval=c(0,Loan))$root)
```

```
## [1] 32622.8
```

Q2

(a)

Base on Question 1, if Scott wants to pay level repayments for this 30-year loan as well, he should pay around 32622.80 each year. However, now he sets up an interest offset account which will increase the total annual repayment by 3500:

```
#Interest offset account balance:
Offset_Acc<-100000
#Interest offset account interest rate (p.a.):
i_off<-0.035
#Interest paid by offset account each year:
(int_repay<-Offset_Acc * i_off)
```

```
## [1] 3500
```

However,the annual loan repayment(excluding int_repay) turns out to be the same as the answer of Question 1:

```
(AnswerQ2a<-AnswerQ1)
```

```
## [1] 32622.8
```

(b)

Now Scott total annual repayment equals to 32622.8 plus the interest produced by the offset account. Therefore he can paid all loan outstanding for less than 30 years. Furthermore, when the loan balance less than 100,000 if we follow the same instalment amount, we should use the whole offset account amount with its annual interest to repay first. The rest of the loan outstanding will be the money I should take out from my wallet. I will write a function to calculate the new loan term:

```
TERM<-function(x,L,i1,A,i2){
  #The five inputs are the annual repayment(x), initia loan amount(L), loan interest rate (i1), interest offset account balance(A), and interest rate of the offset account(i2).
  term<-0
  balance<-L #600000
  int_repay<-A*i2  #3500  #copied Q1's code: interest paid by offset account.
  while(balance>100000){
    balance<-balance*(1+i1) - x - int_repay
    term<-term+1
  }
  return(term)
}
TERM(AnswerQ2a,600000,0.035,100000,0.035)
```

```
## [1] 23
```

The new loan term is 23! Now calculate the last repayment amount. This amount should be not less than 0, and not greater than AnswerQ1 (the original annual repaid amount).
I create a function called "LAST_PAYMENT" to find the last payment:

```
LAST_PAYMENT<-function(x,L,i1,A,i2){
  #The five inputs are the annual repayment(x), initia loan amount(L), loan interest rate (i1), interest offset account balance(A), and interest rate of the offset account(i2).
  balance<-L
  int_repay<-A*i2    #copied Q1's code: interest paid by offset account.
  while(balance>100000){
    balance<-balance*(1+i1) - x - int_repay
  }
  #the variable balance now will be the wrong one since it follows the original instalment amount. Now reverse the calculate to find the loan outstanding at the beginning of the 23th year, and then substracts the offset account balance & its 23th interest from the beginning loan balance. The remaining amount will be the last payment.
```

```
  correct_balance<-(balance + x + int_repay)/(1+i1)
  last<-correct_balance*(1+i1) - 100000 - int_repay
  return(last)
}
LAST_PAYMENT(AnswerQ1,600000,0.035,100000,0.035)
```

```
## [1] 11484.89
```

Use the LAST_PAYMENT function, the last repayment amout is **11484.89**. Note that it is between 0 and 32622.8. That is a reasonable last repayment.
Remark: This amount excludes the 23th interest repayment offset account.

Q3

(a)

Compute the NPV of Theresa's total commission at 19/03/2019 with valuation rate 3.5% p.a.. The level commission payments start from 19/03/2020 to 19/03/2049. The NPV can thus be written as $\$1000 * a_{30}$.

```
i=0.035;v<-1/(1+i)
(AnswerQ3a<-1000 * (1-v^30)/i)
```

```
## [1] 18392.05
```

(b)

Total amount of Bill's loan:

```
options(digits=8)
(Tot_Bill<-600000 + AnswerQ3a)
```

```
## [1] 618392.05
```

This is the present value of the bank's cash outflows at 19/03/2019. We've already calculate the present value of cash inflows at the same date is exactly $600{,}000 = 32622.799\ldots * a_{30}$ with interest rate 3.5%. However, with this rate the PV of inflow and outflow are unequal.

To make this mortgage business break even, the effective IRR should be lower than 3.5%. Numerically, when the interest is set below 3.5%, both PV(inflow) and PV(outflow) will raise due to the increase of a_{30}. However, 32622.8 > 1000 indicates that PV(inflow) will catch up with PV(outflow).

Intuitively, this means that the effective rate of the bank is **lower** than 3.5%, which is used to charge their customers. This makes sense though, since small part of the annual repayment needs to transfer to the broker immediately. The remaining inflows require higher interest rate(3.5%) than the effective rate.

(c)

Analyze the annual cash inflow and outflow amount from the bank's point of view:

```
#Record from 19/03/2019 to 19/03/2049
Inflow<-c(0,rep(AnswerQ1,30))  #use the exact number for annual repayment
Outflow<-c(600000,rep(1000,30))
NPV<-function(rate){
  l<-length(Inflow)
  disc_vector<-1/(1+rate)^(0:(l-1))
  pv<-sum(Inflow*disc_vector)-Tot_Bill #Tot_Bill is from the last chunck.
  return(pv)
}
# Based on my guess in part(a), set the searching interval c(0,0.035)
# The root is the effective rate of the bank with respect to this business.
uniroot(NPV,interval = c(0,0.035))
```

```
## $root
## [1] 0.032602804
##
## $f.root
## [1] 41.851269
##
## $iter
## [1] 3
##
## $init.it
## [1] NA
##
## $estim.prec
## [1] 6.1035156e-05
```

The effective interest rate is thus **3.26%** approximately.
(Remark: Based on the description of the question, assume that the valuation rate for Theresa's commission doesn't change.)

(d)

Suppose his adjusted annual payment is $y when the rate 3.26% is offered. The trick here is the same as Q1:

```
Q3<-function(x){x*(1-(1/(1+0.0326))^30)/0.0326 - 600000}
(New_repay<-uniroot(Q3,interval=c(0,Loan))$root)
```

```
## [1] 31649.27
```

```
(AnswerQ3d<-New_repay - AnswerQ1)
```

```
## [1] -973.52902
```

The new annual instalment becomes $31649.27 approximately which decreases around $ 32622.80 - 31649.27 = $ **$973.53**.
(Remark: Assume the interest rate for Bill is EXACTLY 3.26%.)

Q4

Install the package "lubridate" to help count dates:

```
library(lubridate)

## Warning: package 'lubridate' was built under R version 3.5.3

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date

Birth<-c(ymd("19650516"),ymd("19220301"),ymd("19420921"))
Issued<-c(ymd("19850721"),ymd("19550923"),ymd("19670602"))
Death<-c(ymd("19940622"),ymd("19930721"),ymd("19960429"))
# Function SPAN calculates the time difference between date1 & date2
SPAN<-function(date1, date2){
  Interval<-interval(date1, date2)
  return(as.period(Interval))
}
```

a. Age next birthday at death

This is calculated by taking the year component of the difference between Birth date and Death date (i.e. age last birthday), and plus 1.

```
(Age_a<-SPAN(Birth[1:3],Death[1:3])$year + 1)

## [1] 30 72 54
```

b. Age last birthday at 1 January preceding death

Since it is age last birthday, simply calculate the time difference between birth date and 01/01/(year of death).

```
Record_death<-c(ymd(paste0(as.character(year(Death[1:3])),"0101")))
(Age_b<-SPAN(Birth[1:3],Record_death[1:3]))$year

## [1] 28 70 53
```

c. Age at the birthday in the policy year of death

First step: Find the end date of the policy interval including date of death.
Second step: Calculate the difference between that end date and the Birthday. Adjust 1 year if necessary.

```
Policy_y<-SPAN(Issued[1:3],Death[1:3])$year
#change the Issued month number to a 2-character strings:
#Example: 7 => "07", 12 =>"12"
Policy_m<-c()
for(i in 1:3){
  Policy_m[i]<-ifelse(nchar(as.character(month(Issued[i])))==1,
                 paste0("0",as.character(month(Issued[i]))),
                 as.character(month(Issued[i])))
}

#change the Issued day number to a 2-character strings:
#Example: 7 => "07", 12 =>"12"
Policy_d<-c()
for(i in 1:3){
  Policy_d[i]<-ifelse(nchar(as.character(day(Issued[i])))==1,
                 paste0("0",as.character(day(Issued[i]))),
                 as.character(day(Issued[i])))
}

#Show the end date of the policy year including the date of death
Policy_interval_end<-ymd(paste0(
                 as.character(year(Issued)+Policy_y[1:3]+1),
                 Policy_m,
                 Policy_d))

#Find the answer. This depends on the difference of the policy interval end
and the Birth date.
AnswerQ4c<-c()
for(i in 1:3){
  AnswerQ4c[i]<-ifelse(SPAN(Birth[i],Policy_interval_end[i])$year==
                       year(Policy_interval_end[i])-year(Birth[i]),
                       year(Policy_interval_end[i])-year(Birth[i]),
                       year(Policy_interval_end[i])-year(Birth[i])-1)
}
AnswerQ4c

## [1] 29 71 53
```

The answers are summarized below, correspondingly:
(answer for part a. is in the first column, etc)

```
matrix(c(30,72,54,
         28,70,53,
         29,71,53), nrow = 3)

##      [,1] [,2] [,3]
## [1,]   30   28   29
## [2,]   72   70   71
## [3,]   54   53   53
```